



Le simulateur temps réel TELLURIS pour le langage CORDIS-ANIMA. Modèles physiques, gestes, sons, images

Claude Uhl, Annie Luciani, Claude Cadoz

► To cite this version:

Claude Uhl, Annie Luciani, Claude Cadoz. Le simulateur temps réel TELLURIS pour le langage CORDIS-ANIMA. Modèles physiques, gestes, sons, images. Actes du séminaire du Groupe de travail Animation et Simulation, 1994, Lille, France. pp.1-15. hal-00910610

HAL Id: hal-00910610

<https://hal.science/hal-00910610>

Submitted on 5 Jun 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Le simulateur temps réel TELLURIS pour le langage CORDIS - ANIMA Modèles Physiques, Gestes, Sons, Images

Claude UHL, Annie LUCIANI, Claude CADOZ

ACROE

INPG - 46 av. Félix Viallet - 38031 Grenoble Cedex
Tél : 76 57 46 48 - Fax : 76 57 46 02 - Email : luciani@imag.fr

Résumé :

La machine I.C.I. vise à définir l'ordinateur comme outil d'interface Homme - Machine généralisée par l'introduction d'un mode de communication naturel humain appelé Communication Instrumentale. Dans ce type de communication, l'homme communique avec la tâche via un outil. Cet outil est un objet se comportant comme un objet physique avec lequel l'homme est en relation multi-sensorielle visuelle, acoustique et gestuelle. Pour réaliser ce type de communication entre l'homme et l'ordinateur, nous avons développé un langage de description d'objets physiques (le langage CORDIS-ANIMA), des transducteurs Gestuels à retour d'effort (TGR) et un simulateur temps réel de ces objets produisant en temps réel les sons, les images et les signaux à destination de la perception gestuelle. Nous parlerons plus spécialement de l'architecture matérielle de ce simulateur nommé "Machine TELLURIS"

1. LA MACHINE ICI - Interface de Communication Instrumentale

La machine I.C.I. (Interface de Communication Instrumentale) se rattache au problème général de la Communication Homme/Machine. Il vise au développement théorique, matériel et applicatif du concept de Communication Instrumentale [Cad94]. La Communication Instrumentale est une modalité d'interaction informationnelle qui met en jeu le canal gestuel s'appliquant à un objet physique (instrument au sens large). Lorsque le canal gestuel est exploité selon toutes ses fonctions d'émission d'information (geste sémiotique), d'action (geste ergotique) et de perception (geste épistémique), il est d'une très grande performance pour l'échange informationnel.

L'exemple le plus ancien de communication *instrumentale* est celui de la communication instrumentale musicale.

Nos travaux dans l'image ont fait depuis leur origine l'hypothèse que la révolution informatique dans la production de l'image en mouvement résidait dans l'apparition de l'instrumentalité dans la production d'images [LJF91]. Le contrôle proposé est celui de contrôle gestuel. C'est le même concept qui émerge aujourd'hui avec le terme de "performance animation".

Cette situation instrumentale est instanciée dans le contexte de l'ordinateur en développant des simulations d'objets physiques avec lesquels un opérateur peut interagir de manière multi-sensorielle (gestuelle, acoustique, visuelle et tactilo-kinesthésique) et en temps réel.

Elle est d'autre part élargie et généralisée à d'autres finalités que la production d'événements sonores ou visuels. La méthode met en avant le concept d'objet ou "instrument" de communication qui pourrait être vu comme une généralisation de l'icône du Macintosh, en ce que cet objet, qui est l'intermédiaire des processus de communication, n'est pas seulement un graphisme métaphorique, mais un objet intégral, c'est-à-dire intrinsèquement multi-sensoriel, et manipulable gestuellement de manière élaborée.

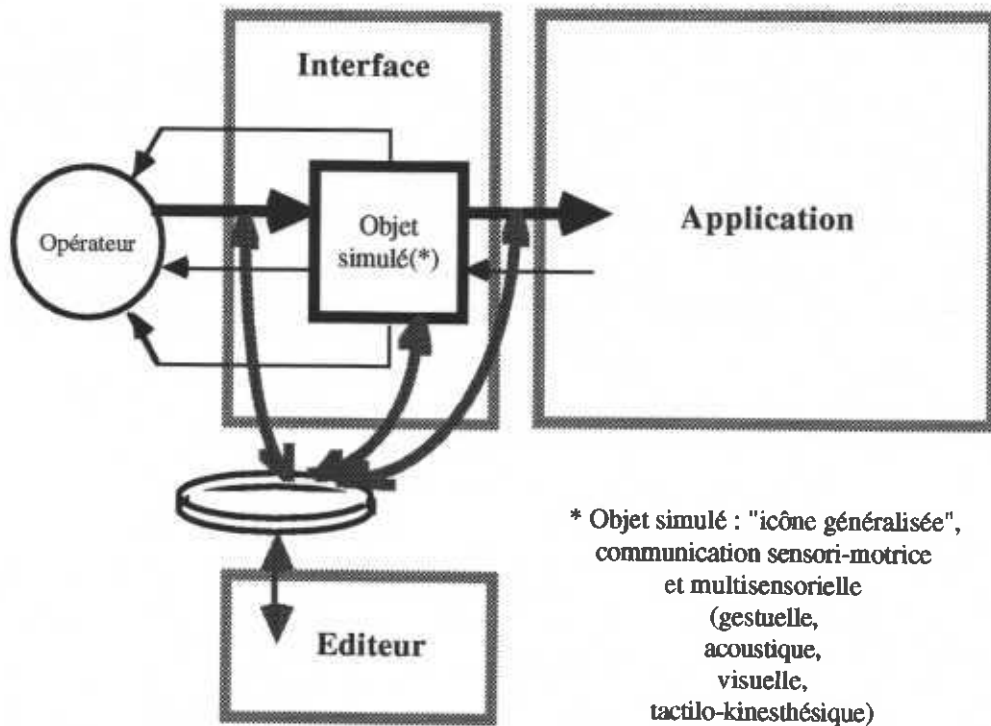
Le principe de la communication instrumentale avec l'ordinateur est basé sur la simulation d'un objet physique par ce dernier, de telle façon que toutes les déterminations de cette relation soient prises en compte. L'opérateur, pour la partie de la maîtrise de la tâche qui peut ainsi se formaliser, est ainsi en interaction avec un objet (virtuel), dont les caractéristiques et les propriétés auront été définies au préalable, à l'issue d'une analyse et d'une modélisation de la situation. Il peut alors s'établir un processus d'interaction entre l'opérateur et l'objet pendant lequel ils s'échangent information et énergie [Luc93].

Dans la transposition informatique de cette situation, nous avons donc à considérer plusieurs problèmes spécifiques :

1. la constitution d'organes d'interface qui autorisent la communication gestuelle selon tous ces aspects (les TGR). Ce premier point constitue un premier axe des travaux de l'ACROE ainsi que du programme I.C.I. ;
2. la mise en œuvre de programmes de calcul capables de recevoir les informations gestuelles captées par les TGR et, partant de là, de produire des signaux (cohérents entre eux), à destination des interfaces (transducteurs) acoustiques, visuels et en retour au gestuel (tactile).
3. Les algorithmes de simulation par modèles physiques, base du système CORDIS-ANIMA, nécessitent des performances et des conditions de structure et d'architecture particulières pour les machines qui vont les exécuter. La conception et la mise en œuvre des conditions matérielles de cette simulation constituent un troisième axe des travaux .

I.C.I.

INTERFACE DE COMMUNICATION INSTRUMENTALE



Simulation d'objets physiques supports de la communication
Langage de définition de ces objets

Dialogue avec l'application : sous la forme de la manipulation
d'un objet physique simulé (icône généralisée)

Communication de l'information extraite vers l'application
Retour de l'information de l'application vers l'objet

Traitement et édition de l'information de communication

2. LES TRANSDUCTEURS GESTUELS RETROACTIFS - RAPPELS

La relation instrumentale s'appuie sur trois modalités sensorielles combinées et consécutives à une action gestuelle : une perception sonore, une perception visuelle et une perception proprio-tactilo-kinesthésique, appelée par la suite en abrégé dans le texte perception gestuelle. Pour les deux premières modalités, des transducteurs classiques (écran et haut parleur) existent. Pour le canal de perception gestuelle, des transducteurs spécifiques doivent être développés, appelés TGR - Transducteurs Gestuels Rétroactifs. Ceux-ci doivent assurer la double fonction d'émission des données mécaniques du geste instrumental (fonction de capture) et reproduire des phénomènes (déplacement, force) concernant la perception gestuelle.

Notre équipe a développé plusieurs prototypes successifs de TGR : un système 1 degré de liberté DDL sur 50 cm de déplacement en 1978, un système à 1 DDL très compact en 1980, 1 système très compact à 16 degrés de liberté en 1988 [CLF88]. Ce dernier dispositif est basé sur un module actionneur capteur très compact, développé dans notre équipe, appelé "actionneur modulaire plat". Ces modules s'assemblent linéiquement pour former le dispositif, nommé "Clavier Rétroactif Modulaire", ayant le nombre de degrés de liberté voulus.

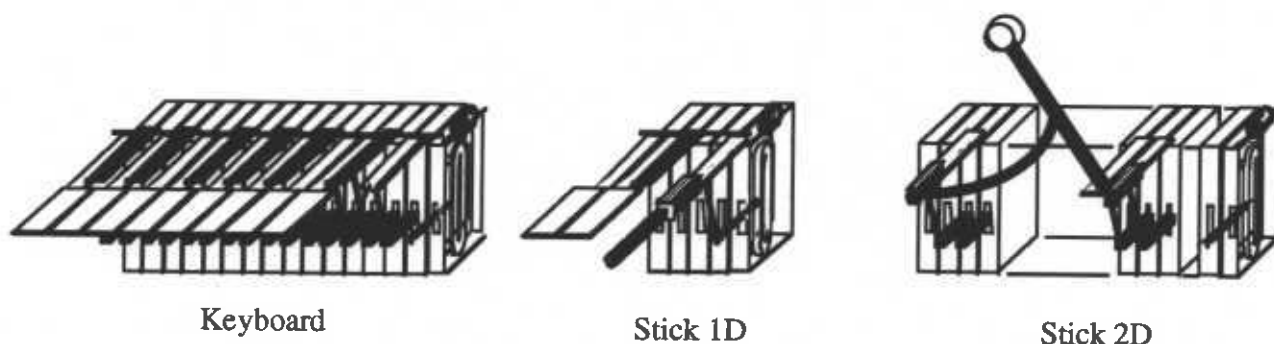


Figure 1.

L'originalité de ce dispositif réside dans les performances dynamiques et énergétiques qu'il peut développer dans un encombrement réduit. Ces dimensions sont celles d'une touche de piano et approximativement celles d'un clavier alphanumérique standard. Il permet de restituer la sensation de chocs très durs ou de frictions fines sur une surface. En plus de ses bonnes performances dynamiques, le clavier rétroactif modulaire peut s'adapter aux différentes morphologies du geste de manipulation. En plus de la configuration type piano, différents mécanismes d'habillage du clavier ont été développés. Ce sont des accessoires mécaniques qui se greffent sur les touches du clavier rétroactif modulaire réalisant ainsi des dispositifs de manipulation gestuels à deux ou à trois degrés de liberté, voire même plus, par l'association mécanique de touches indépendantes à un degré de liberté du clavier rétroactif modulaire.

3. CORDIS ANIMA

Cordis - Anima est un système de modélisation et de simulation numérique en temps réel d'objets physiques [CLF90]. Autrement dit, Cordis - Anima est d'une part un mode de représentation des objets physiques de l'univers, et d'autre part, un outil de calcul de systèmes physiques et de production de phénomènes audibles, visibles et manipulables.

Il a donc fallu formaliser les objets physiques par une représentation discrète et de dimension finie [CLF91]. Les plus petits objets physiques de Cordis - Anima sont les éléments <MAT> et <LIA>, qui constituent le cadre formel des algorithmes.

<MAT>: **point matériel**, produit une position à partir d'une force, le point pouvant être dans un espace à 1, 2, 3,... dimensions ; de même, la force peut être uni, bi, tri... dimensionnelle.

<LIA> : **élément de liaison**, produit deux forces opposées de même intensité à partir de deux positions.

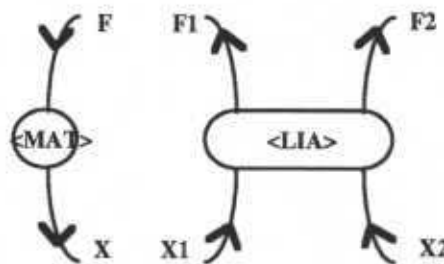


Figure 2.

A l'aide des éléments <MAT> et <LIA>, tous les objets CORDIS peuvent être représentés comme des réseaux, dit **Réseaux <MAT-LIA>**.

Les algorithmes se rapportant aux éléments MAT et LIA sont issus des lois de la physique. Ainsi, l'algorithme fondamental d'une masse (module MAT) est donné par la relation fondamentale de la dynamique : $F = M \cdot \Gamma$. Ensuite par une approximation simple, tout en tendant vers la Relation Fondamentale de la Dynamique, on obtient l'algorithme de la physique discrète : $X(n) \leftarrow -2 \cdot X(n-1) + X(n-2) + 1/M \cdot F(n-1)$.

On obtient par le même type d'approximations, les algorithmes se rapportant aux modules LIA, représentant des interactions de type ressort ou frottement.

Les non linéarités mécaniques sont représentées par une logique conditionnelle à l'intérieur des modules <LIA>.

4. LE SIMULATEUR TELLURIS

Le projet Telluris est partie intégrante du projet ICI (Interface de Communication Instrumentale), et consiste à définir une architecture de machine performante pour la simulation d'objets physiques en temps réel. Développé par l'ACROE, Cordis Anima est un mode de représentation des objets physiques de l'univers, visant à les simuler par le son, l'image et le geste. Cette représentation est possible en temps réel sous certaines contraintes. En effet, si l'opérateur désire visualiser, entendre et ressentir l'effort qu'il a produit de manière instantanée, il faut que le délai inévitable dû au temps de calcul soit imperceptible, donc inférieur à une durée donnée. Cette durée est d'une milliseconde pour l'image et le geste (opérations cadencées à 1 kHz), et de 22,7 microsecondes pour le son (opérations cadencées à 44 kHz).

Dans le cadre du projet Telluris, nous avons d'abord étudié la solution utilisant des processeurs RISC I860 [JN92] qui ne nous a pas satisfaite [Wi92]. Puis, nous avons chiffré la puissance nécessaire à la simulation de trois modèles typiques : le piano, l'agglomérat et le véhicule planétaire [Uhl93]. Les résultats obtenus montrent qu'un bon rendu des objets simulés nécessiterait d'effectuer entre 500 et 1000 millions d'opérations flottantes en une seconde. Compte tenu des caractéristiques actuelles des machines existantes [GB92], ces performances ne peuvent être atteintes par l'utilisation d'un processeur seul, ce qui nous conduit inévitablement au parallélisme [HP91].

5. CONTRAINTES APPORTÉES PAR LA MODÉLISATION CORDIS ANIMA

5.1. Contraintes apportées au simulateur

La simulation d'un objet physique dans le temps consiste à modifier le réseau d'interconnexion entre les composants de l'objet (modules MAT) et à calculer les algorithmes numériques associés aux points de communication pour connaître les nouvelles positions et forces internes à l'objet.

Le simulateur est alors la machine numérique qui a pour rôle d'exécuter les algorithmes numériques du modèle simulé, et la qualité de la simulation dépendra des performances de la machine. Elle comporte des programmes, des mémoires, une horloge qui va conditionner tout le bon fonctionnement de la machine. Elle est reliée au monde extérieur par des convertisseurs : Analogiques Numériques pour lui transmettre des données à traiter et Numériques Analogiques pour retransmettre les résultats du traitement au monde extérieur. Les données numériques en entrée et en sortie sont des signaux échantillonnés obéissant aux conditions de Shannon, ce qui signifie que le signal ne contient pas de composantes à des fréquences supérieures ou égales à f_{max} , et est donc entièrement déterminé par la suite de ses valeurs à des instants régulièrement espacés de la durée $T = 1/2f_{max}$.

5.2. Contraintes apportées à la relation au monde extérieur, à l'opérateur

Un modèle Cordis Anima résulte de l'agencement selon une manière particulière de modules élémentaires (réseau MAT LIA). Un certain nombre de ces modules peuvent être reliés au monde extérieur, c'est à dire à l'opérateur.

Celui-ci peut donc agir sur l'objet en modifiant une variable d'entrée (il peut tout aussi bien s'agir d'une force que d'une position). Ainsi, si l'objet est au repos à l'état initial, cette modification d'une variable d'entrée va induire une modification du réseau d'interconnexion d'éléments MAT et LIA représentant l'objet physique, que l'opérateur veut pouvoir observer en temps réel.

Les éléments extérieurs auxquels la machine est connectée et doit transmettre les résultats à chaque période T sont les suivants : les transducteurs gestuels à retour d'effort (appelés TGR), le haut-parleur, l'Écran.

A l'instant nT , il y a arrivée d'une donnée ou d'un flot de données de type {position}, depuis les TGR. Le simulateur exécute les algorithmes et, à l'instant $(n+1)T$, renvoie une donnée ou un flot de données de sortie de type {force} aux TGR (retour d'effort), et une donnée ou un flot de données de type {position} à l'écran. D'autre part, les données ou flots de données de déformations acoustiques, calculés à 44 kHz sont transmis par paquets de 44 échantillons de manière synchrone à 1 kHz vers le système de sortie sonore numérique qui les restitue à 44 kHz sur les haut-parleurs.

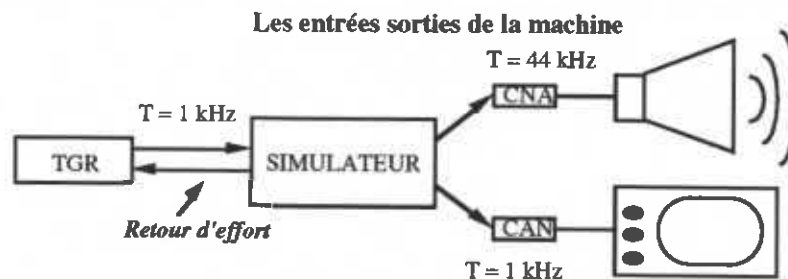


Figure 3.

Récapitulatif des besoins chiffrés en entrée/sortie :

Besoins gestuels : 64 touches de Transducteur Gestuel, codées sur 16 bits à une fréquence de 1 kHz, représentent un débit de 130 Ko/s en unidirectionnel, et 260 Ko/s en bidirectionnel.

Besoins sonores : 4 canaux de sortie codés sur 18 bits, à 44,1 kHz, représentent un débit de 350 Ko/s.

Besoins images : 1000 points à trois coordonnées sur 32 bits à un taux de balayage de 32 fois par seconde, représentent un débit de 380 Ko/s.

Tous accès concurrents, on obtient 952 octets/ms, et si on réserve 90% du temps au calcul, on aboutit finalement à 9,5 Mo/s.

5.3. La T-Simulabilité

Les instants d'entrée et les instants de sortie doivent être synchrones, et entre les entrées et les sorties qui en résultent, il doit s'écouler au pire une seule période T . Etant donné un calculateur qui demande un temps T_c de calcul entre une entrée de l'extérieur (par exemple de TGR) et une sortie vers l'extérieur (par exemple le TGR). Pour que le modèle soit correct vu de la personne qui manipule l'objet virtuel via les senseurs et actionneurs du TGR (correct veut dire qu'il s'agit bien d'un simulateur d'un objet réel donné, voulu), il faut globalement que la bande passante de cet objet soit inférieure ou égale à T_c . Ces objets seront dit T-simulables. Dans le cas inverse, ils seront dits non T-simulables. Si l'objet est seul, en fonction du principe de synchronisme entre les sorties,

une sortie ne peut être influencée par une sortie qu'avec un retard T . Aussi, bien que les variables d'entrée et de sortie soient correctement discrétisées relativement aux conditions du théorème de Shannon, un décalage temporel peut apparaître entre le moment où l'algorithme produit un événement en sortie et celui où, selon son modèle, cet événement aurait dû se produire. On considère alors deux types d'objets : ceux pour lesquels l'influence d'une variation à l'entrée ne se fait jamais sentir en sortie dans un délai inférieur à T , appelés objets **T-Simulables**, et ceux pour lesquels aucune valeur de T n'est assez petite pour obtenir cette situation, appelés objets **non T-Simulables**. Les dérives produites par la non T-simulabilité ne correspondent pas à des approximations plus ou moins bonnes de l'objet physique, mais à un modèle dynamiquement faux. Il est ainsi impossible de simuler un ressort $F=-K.x$, connecté sur l'extérieur puisque $F(t)=K.x(t)$ se transforme nécessairement en $F(t+Dt)=K.x(t)$ et l'opérateur ne manipulera pas un ressort mais un ressort spécial "ressort-ligne à retard". La question de la T-simulabilité vient du fait que l'objet naturel est remplacé par un algorithme où l'on transforme en opérations arithmétiques donc séquentielles au moins localement, des opérations qui ne le sont pas. On pourrait dire que l'objet naturel (le ressort vrai par exemple) a une puissance de calcul infinie, ce qui n'est pas le cas d'une machine à calculer numérique.

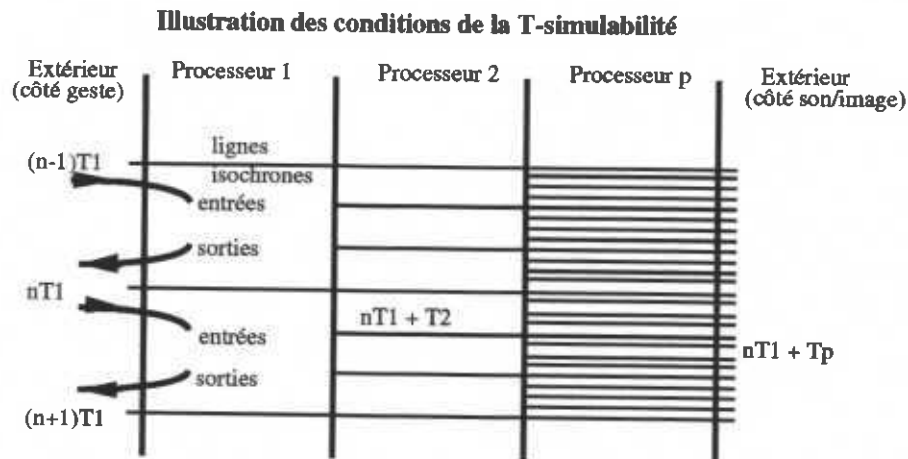


Figure 4.

6. LES MODELES EVALUATEURS

Dans la recherche d'un simulateur efficace, il faut, lors de tests de différentes machines, disposer de modèles typiques regroupant les principales difficultés liées à ce type de simulation.

6.1. le piano

Le piano est un modèle petit et homogène, mais simulé à haute fréquence.

Dans la modélisation CORDIS ANIMA, un piano est constitué d'un ensemble de cordes couplées entre elles par des liaisons de type visco-élastique. Il prend en compte les couplages entre cordes par la table. Le couplage physique entre les éléments de cette structure, pour être simulé, nécessite des capacités de communication inter processeurs élevées avec une fréquence d'échange des données de 44 kHz.



Nombre d'opérations pour une corde à M Masses : $6 + 12 \cdot M$ Opérations à 44 kHz

Selon la machine, il faudra évaluer qui des accès mémoire ou des opérations, constituent le facteur limitant.

L'agglomérat est un modèle homogène et fonctionnant à basse fréquence, mais très volumineux.



Le modèle, totalement homogène, nécessite peu de paramètres. Néanmoins, on va dénombrer les opérations à effectuer et les accès mémoire pour un agglomérat à M masses.

Nombre d'opérations : $13*M^2+6*M$ Opérations en 1 ms

Nombre d'accès mémoire : $10*M^2$ lectures et $3*M^2$ écritures en 1 ms.

6.3. Le véhicule planétaire

Un véhicule est un objet structuré se déplaçant sur un sol fixe, mobile ou déformable de type agglomérat, avec lequel une partie du véhicule (les roues) est en interaction. Nos études ont montré que ce cas se ramène alors au modèle précédent, avec la différence de l'inhomogénéité du modèle, en terme de nombre de paramètres et d'interactions différentes : si M masses entrent en interaction avec P masses, l'algorithme sera de complexité :

$$M*\frac{M-1}{2} + P*\frac{P-1}{2} + M*P$$

6.4. Conclusion

Récapitulatif des besoins chiffrés en puissance pour les modèles typiques piano et agglomérat :

La puissance nécessaire en MFLOPS peut être d'autant plus grande que l'on a d'interactions entre les processeurs à 44 kHz. En effet, un modèle sonore devra être T-Simulable à 44 kHz, tandis que la plupart des autres modèles le seront à 1 kHz. Le nombre de calculs à effectuer dépend de la quantité de connexion des points de communication.

Piano : le modèle composé de 64 cordes à 20 masses couplées en anneau à 44 kHz, comporte 16200 opérations à effectuer en 22,7 ms, ce qui correspond à 720 MFLOPS.

Agglomérat : avec 200 masses en interaction totale, le modèle calculé à 1 kHz nécessite 521200 opérations à effectuer en 1 ms, ce qui correspond à 521 MFLOPS, et avec 300 masses on atteint déjà le GFLOPS.

Capacité mémoire : la capacité mémoire nécessaire dépend du modèle par sa grosseur et son homogénéité et de la structure de la machine. Un gros modèle est celui de l'agglomérat à 300 masses, qui implanté sur plusieurs processeurs nécessite de recopier l'ensemble des données dans chaque mémoire. Dans le cas d'un réseau distribué : par processeur, on compte alors 380 Ko de mémoire. D'autre part, on évalue le nombre de données différentes qu'un processeur rapide, TFP (SGI) peut traiter à 1 kHz : deux accès mémoire peuvent être effectués par cycle d'horloge à 75 MHz, ce qui équivaut à 1,2 Mo de mémoire.

Trouver le bon compromis

Il s'agira de trouver la meilleure architecture permettant d'être appliquée à tous les modèles, sans qu'une contrainte ne soit du tout solutionnée. Il faut en particulier, que les contraintes des modèles 1 et 2 (corde et agglomérat) soient totalement résolus. Le troisième modèle, le véhicule planétaire, pourra ensuite être testé, en particulier pour l'inhomogénéité et le grand nombre de paramètres différents.

Aucun choix définitif ne devra être fait avant de s'être assuré que ces trois modèles seront aisément simulés.

7. CARACTERISTIQUES DU SIMULATEUR

L'étude du parallélisme comprend souvent 2 approches, la première étant d'utiliser un grand nombre de processeurs ne présentant pas de puissance particulièrement, la quantité permettant d'obtenir les performances voulues. La deuxième possibilité est de répartir l'application parallèle sur peu de processeurs (de 2 à 8), chacun d'eux étant très performant. Compte tenu du haut degré d'interconnectivité entre modules dans la plupart des réseaux MAT et LIA représentant un objet physique, la première solution ne peut être retenue et l'on sait aujourd'hui que seul des processeurs dont les performances sont déjà très élevées fonctionnant en uniprocasseur peuvent solutionner la problématique de simulation temps réel.

7.1. Spécificité du processeur

Les algorithmes Cordis Anima font intervenir différents types de données à fréquence variable au cours d'une période de simulation. En effet, on ne considère pas de la même manière, paramètres, instructions, adresses des données variables et données {forces} et {positions}, qui varient au cours de la simulation. Ces deux variables sont elles-mêmes de type différent. Les paramètres et instructions, invariables tout au long de la simulation d'un modèle donné, ne sont pas vus de l'extérieur, sauf lors de l'initialisation, ou chargement du modèle. Les adresses également invariables sont transmises lors du couplage d'une masse avec une masse présente sur un autre processeur. Enfin, les variables seront lues, modifiées : une variable {force} est lue une fois et modifiée une fois par période de simulation, une variable {position} est lue deux fois et modifiée une fois par période de simulation. Puis certaines de ces variables sont échangées d'un processeur à un autre

Pour satisfaire la spécificité de chaque type de données, l'idéal serait de disposer de plusieurs mémoires indépendantes auxquelles les différentes unités pourraient avoir accès simultanément. Les données utilisées le plus fréquemment par un processeur sont bien évidemment les instructions. Un cache instructions serait assez approprié, mais il doit être suffisamment grand afin de ne générer aucun défaut cache lors de l'exécution du programme.

Aussi, on en déduit les règles suivantes dans une architecture multiprocesseurs :

- les données non partagées sont les paramètres, les instructions, les adresses des données, les positions des masses internes calculées à $T-2$, les forces résultant des algorithmes des liaisons connectant les masses internes du processeur.
 - les données partagées par deux processeurs sont les forces résultant des algorithmes de liaison connectant les masses du premier processeur à celles du deuxième.
 - les données partagées par plus de deux processeurs sont les données positions résultant des algorithmes masses.
- Dans le cas du modèle du piano avec couplage en anneau, aucune donnée n'est partagée par plus de deux processeurs. Pour le modèle de l'agglomérat, les données positions sont partagées par tous les processeurs. Enfin, dans le cas du modèle du véhicule, les positions des masses sol sont partagées par tous les processeurs (car

couplées à toutes les autres masses du modèle), tandis que selon la répartition, les autres données positions seront peu ou ne seront pas partagées.

Reprenons l'exemple des trois modèles typiques. Pour le modèle du piano, chaque masse d'une corde est connectée à deux liaisons, sauf la masse de couplage (à l'extrémité), reliée en plus à deux liaisons de couplage. Pour le modèle de l'agglomérat à M masses, chaque masse est reliée à $(M-1)$ liaisons. Pour le véhicule, les masses sont connectées à plus ou moins de liaisons, selon qu'elles appartiennent au châssis, à une roue ou au sol.

On s'efforcera donc de répartir les données dans différentes mémoires de manière exploiter au mieux le typage des données.

7.2. Les communications

Les algorithmes décrivant les objets sont totalement synchrones. Une définition du synchronisme dans le cadre d'applications parallèles est donnée par Tsitsiklis et Bertsekas [BTs] :

Supposons qu'une séquence de calcul ou d'opérations soit divisée en segments consécutifs, appelés "phases". Les calculs effectués à chaque phase sont partagés entre les n processeurs du système. Durant une phase t , chaque processeur i effectue des calculs avec les données disponibles en mémoire, ainsi que les informations reçues des autres processeurs aux phases antérieures. Puis chaque processeur i envoie des informations aux autres processeurs. La phase suivante peut commencer lorsque tous les messages ont été transmis. Ce que l'on veut pouvoir arriver à réduire, ce sont les temps de pénalisation induits par les communications.

Le synchronisme, tel qu'il est décrit ici, est assez pénalisant, car il faut attendre que tous les processeurs aient fini leur phase de calcul pour la mise à jour des données. C'est ce problème qui se pose pour les algorithmes Cordis Anima.

Un autre aspect à étudier est le phasage des communications dans la simulation d'un modèle Cordis Anima. Au jour d'aujourd'hui, en temps réel, les communications sont monophasées (Figure 7.): la durée de la phase de communications est la même pour chaque processeur. C'est ce qui va limiter le nombre maximum de masses que l'on va pouvoir simuler (par modèle), et qui correspond, selon la répartition sur les processeurs, au nombre d'algorithmes MAT et LIA simulables dans la durée T privé du temps nécessaire aux communications.

Deux autres possibilités sont envisageables et à étudier : les communications biphasées (Figure 8.), où l'on sépare une période T en deux sous-périodes, de durée τ et $T-\tau$, τ n'étant pas de durée fixe et prédéfinie, mais variant selon le modèle et la répartition. Cette solution permet de toujours travailler sur les données effectives, c'est à dire mises à jour : en effet, dans le premier cas, l'algorithme MAT utilise une force calculée au cycle antérieur, qui n'est donc pas la force mise à jour au cycle courant. La deuxième possibilité est l'asynchronisme des communications : seules les exécutions des algorithmes sont strictement cadencées lors des cycles de période T . Les communications de toutes les données sont déclenchées dès qu'un certain nombre de données sont prêtes. Des données sont utilisées par les processeurs destinataires dès leur réception, ce qui provoquera des aléas et une non maîtrise des données à chaque instant. Ce dernier cas devrait plutôt être étudié expérimentalement, une fois la

machine choisie, pour juger à la fois de la validité de la modélisation, et de l'impact sur la machine (à voir en terme de tests pratiques sur machine).

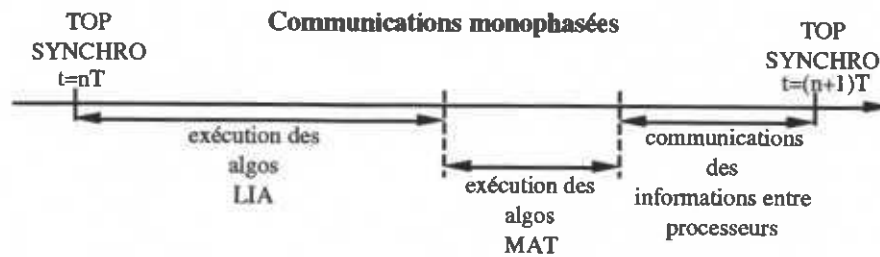


Figure 7.

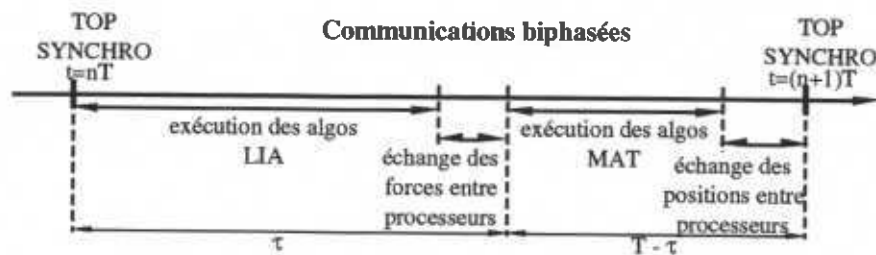


Figure 8.

Lorsque l'on implante un modèle Cordis-Anima sur plusieurs processeurs [BYA89], on peut concevoir que l'on effectuera les communications inter processeurs de manière différente selon le type de la donnée : au cours de la période de simulation T (à 1kHz pour l'image, 44 kHz pour le son), un processeur va utiliser une ou plusieurs données position importées depuis d'autres processeurs et calculées à $T-1$. Cette donnée position doit rester inchangée dans le processeur hôte, pendant toute la période T . Or c'est pendant la durée de cette période que le processeur propriétaire de la donnée va en modifier la valeur. Aussi, la transmission de la nouvelle position ne peut être effectuée avant la fin de la phase de calcul. Par contre, lorsqu'une donnée force a été utilisée dans le calcul d'un algorithme masse, elle peut être modifiée, car elle n'est utilisée qu'une seule fois. Le transfert d'une donnée force une fois sa nouvelle valeur calculée, peut donc se faire immédiatement sans attendre la fin de la phase de calcul des LIA.

Éliminée jusqu'à présent, la solution des mémoires partagées peut être envisagée, depuis l'avènement de processeurs très performants (4 opérations par cycle à 75 MHz pour TFP de SGI), disposant de gros caches et d'une mémoire globale. Elle est particulièrement adaptée aux modèles comportant des données {positions} partagées par plus de deux processeurs, comme l'agglomérat et le véhicule planétaire. Notons que dans le cas de l'agglomérat, les données {positions} calculées par un processeur, sont utilisées par tous les autres processeurs. La référence à une mémoire centrale, commune à tous les processeurs, peut alors se révéler très coûteuse lorsque le nombre de processeurs est important. Le processeur seul doit être suffisamment performant, pour que les performances nous satisfassent avec peu de processeurs (de 2 à 8 maximum). Par contre, lorsque les données {positions} sont partagées par deux processeurs au plus (le piano), le mode de communication en anneau sur processeurs à mémoire distribuée serait plus adapté. Dans ce cas, chaque processeur ne communique qu'avec ses deux voisins, il n'y a donc pas de bus commun à gérer. Tous les processeurs communiquent pendant toute la

durée de la phase de communication : les communications entre processeurs se feront par bloc mémoire les plus gros possibles (pour diminuer le temps d'overhead). Les communications se font par transfert de données de mémoire à mémoire (si possible statique, les plus rapides). Il est important de pouvoir instancier le type de communication en anneau avec une durée de communication constante quelque soit le nombre de processeurs.

8. ÉTUDE DE CAS

Nos études se sont basées sur des tests effectués sur divers processeurs [DB91] : un processeur RISC (INTEL I860) [JN92] [INT89], un DSP (Texas Instrument TMS320C40) [TI92], un processeur VLIW -Very Large Instruction Word- (INTEL iWarp) [Bork90], un CRAY (J916) ainsi qu'un processeur spécifique (Silicon Graphics TFP).

On donne ci-dessous, un tableau et une courbe détaillant les résultats obtenus :

Récapitulatif des résultats obtenus à ce jour
Nombre de masses simulables pour les 2 cas typiques

Processeurs	AGGLOMERAT	PIANO
AP120 1 proc. 1 proc.	10 R 16 T	12 T
IWARP 1 proc.	25 T	17 R
I860 1 proc. 8 proc.	44 T 140 T	32 T 256 T
DSP C40 1 proc. 8 proc.	40 T 102 T	38 T 304 T
TFP SGI 1 proc.	76 R	176 T
CRAY J916 1 proc. 4 proc.	100 R 150 R	300 R 900 R

R : résultats réels, obtenus lors de benchs matériels.

T : résultats théoriques, déduits des performances théoriques, ou des résultats réels dans un autre contexte.

Figure 9.

Les machines utilisées actuellement sont les AP120 et AQ120. Leurs performances sont de 2x12 MFLOPS peak (pointes), et de 2x6 MFLOPS sustain en utilisation réelle, soit 12 MFLOPS. L'objectif serait d'atteindre au moins 120 MFLOPS sustain, soit 400 à 500 MFLOPS peak. Est-ce réalisable à l'heure actuelle ? Il semblerait que oui, d'après les résultats annoncés par Cray et Silicon Graphics.

9. BIBLIOGRAPHIE

- [CLF90] C. Cadoz, A. Luciani, J-L Florens,
CORDIS - ANIMA : système de modélisation et de simulation d'instruments et d'Objets Physiques pour la Création Musicale et l'image animée - Actes Colloque "Modèles Physiques, Création Musicale, Ordinateur", Grenoble 1990, Editions de la Maison des Sciences de l'Homme
- [CLF91] C. Cadoz, A. Luciani, J-L Florens & O. Raoult
Physique Discrète: Discrétisation du temps et de la matière - Colloque de Géométrie Discrète - Grenoble - 1992
- [CLF88] Claude Cadoz, Leszek Lisowski, Jean Loup Florens,
Clavier rétroactif modulaire et actionneur modulaire plat - Brevet Français. Déposé le 13 / 10 / 88. N°88 14064. Extension International 1989.
- [LJF91] Annie Luciani, Stéphane Jimenez, Jean Loup Florens, Olivier Raoult-1991 -
Computational physics : a modeler simulator for animated physical objects - Proceedings of the European Computer Graphics Conference and Exhibition. Vienna, Austria, Septembre 91, Editeur Elsevier
- [Luc93] Annie LUCIANI - 1993
The concept of the Instrumental Communication Interface : Towards a complete representation by means of computer - 1er Colloque franco-japonais sur les environnements synthétiques - Japon - Dec 93 - A paraître Springer Verlag Ed.
- [Cad94] Claude CADOZ - 1994
Le geste, canal de communication Homme / Machine : La communication instrumentale - Numéro spécial de TSI "Interfaces Homme/Machine - Vol 13, n°1, 1994 - Hermès Ed.
- [JN92] P. Joudrier, Y. NGuyen
Cartes Multiprocesseurs Architecturées autour de 1860 pour la Simulation Multisensorielle - Rapport de projet 3ème Année ENSERG - ENSIMAG - Juin 92
- [Uhl93] C. Uhl,
Architecture de machine pour la Simulation d'objets physiques, DEA SIP - INPG, Sept 93
- [BTs] I.N. Bertsekas, D.P. Tsitsiklis
Parallel and Distributed Computation - , Numerical methods, Prentice Hall
- [INT89] INTEL
1860 64 bit Microprocessor advanced Information.
- [HP91] J.L. Hennessy, D.A. Patterson,
Architecture des ordinateurs : Une approche quantitative. - McGRAW-HILL
- [TI92] TEXAS INSTRUMENT
TMS320C4x User's Guide
- [GB92] Gordon BELL
Ultracomputers : A Teraflop Before its Time, Communications of the ACM, August 92
- [BYA89] Laxmi N. Bhuyan, Qing Yang, Dharma P. Agrawal,
Performance of multiprocessor interconnection networks, IEEE Computer, February 1989
- [Bork90] S. Borkar, R Cohn, G. Cox, T. Gross, H.T. Kung, M. Lam, M. Levine,
ProSupporting Systolic and Memory Communication in iWarp, 17th Annual Symposium on Computer Architecture, Mai 1990
- [DB91] David H. Bailey,
Twelve ways to fool the masses when giving performance results on parallel computers. Supercomputer, September 91
- [Wi92] Dave Wilson
Is RISC or DSP best for your application, Computer design, April 1992